PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-056787
(43)Date of publication of application : 03.03.1995

(51)Int.Cl. G06F 12/00

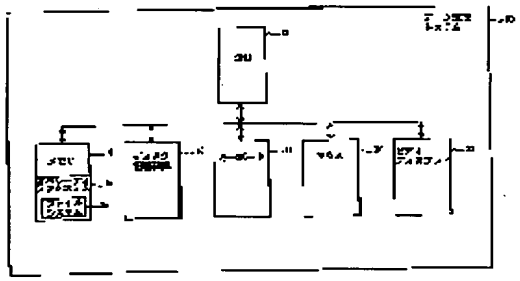
(21)Application number : 06-141923
(22)Date of filing : 23.06.1994

(71)Applicant : MICROSOFT CORP
(72)Inventor : ZBIKOWSKI MARK
BERKOWITZ BRIAN T
FERGUSON ROBERT I

(30)Priority
Priority number : 93 85483 Priority date : 30.06.1993 Priority country : US

(54) METADATA STRUCTURE AND METHOD FOR DEALING WITH THE SAME

(57)Abstract:
PURPOSE: To provide a file system for storing both data and metadata in a disk as the group of flows.
CONSTITUTION: A file system 26 stores data and metadata in a similar configuration. The lowest level of file data stored in a disk 16 is a flow constituting logically adjacent byte groups. The related flows seen in a file, directory, or sub-directory are stored in a node data structure in a variable size. The node data structure in the variable size is stored in the array of the fixed size packet of a disk space. The related node data structure is stored in a catalog data structure. The catalog data structure is stored in the array of the fixed size packet of a disk space. Next, the packet array of the fixed size packet of the disk space is stored as the flows.



LEGAL STATUS

[Date of request for examination] 20.06.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平7－56787

(43)公開日　平成７年(1995)３月３日

(51)Int.Cl.[6]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ０６Ｆ　12/00	５２０　Ｊ	8944－5B		
	Ｅ	8944－5B		

審査請求　未請求　請求項の数29　ＯＬ　（全 12 頁）

(21)出願番号　特願平6－141923

(22)出願日　平成６年(1994)６月23日

(31)優先権主張番号　０８／０８５４８３

(32)優先日　1993年６月30日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　391055933
マイクロソフト　コーポレイション
ＭＩＣＲＯＳＯＦＴ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　ワシントン州　98052－6399　レッドモンド　ワン　マイクロソフト　ウェイ　（番地なし）

(72)発明者　マーク　ズビコウスキィ
アメリカ合衆国　ワシントン州　98072　ウッディンヴィル　ノース　イースト　ワンハンドレッドアンドセヴンティエイス　プレイス　15817

(74)代理人　弁理士　中村　稔　（外７名）

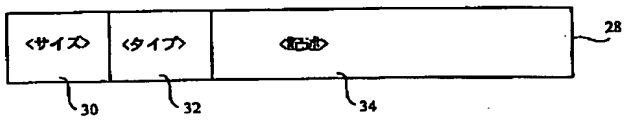
最終頁に続く

(54)【発明の名称】　メタデータ構造体及びその取り扱い方法

(57)【要約】

【目的】　データ及びメタデータの両方をディスクに流れのグループとして記憶するファイルシステムを提供する。

【構成】　ファイルシステムはデータ及びメタデータを同様の形態で記憶する。ディスクに記憶されたファイルデータの最低のレベルは、論理的に隣接するバイトグループを構成する流れである。ファイル、ディレクトリー又はサブディレクトリーに見られる関連する流れは、可変サイズのオノードデータ構造体に記憶される。可変サイズのオノードデータ構造体は、ディスクスペースの固定サイズバケットのアレーに記憶される。関連オノードデータ構造体はカタログデータ構造体内に記憶される。カタログデータ構造体は、ディスクスペースの固定サイズバケットのアレー内に記憶される。次いで、ディスクスペースの固定サイズバケットのバケットアレーが流れとして記憶される。



【特許請求の範囲】

【請求項１】　ディスク記憶装置と、オペレーティングシステムを実行する処理手段とを有するデータ処理システムであって、

（ａ）ディスク記憶装置において論理的に隣接するデータタイプを保持するための第１の可変サイズの流れデータ構造でデータを記憶し、

（ｂ）ディスク記憶装置において論理的に隣接するデータバイトを保持するための第２の可変サイズの流れデータ構造でメタデータを記憶し、そして

（ｃ）上記流れデータ構造の各々に対し、その流れデータ構造がディスク記憶装置のディスクにいかに記憶されるかを識別するタイプ識別子を含む流れ記述子を記憶する、という段階を備えたことを特徴とする方法。

【請求項２】　メタデータをディスク記憶装置において第２の可変サイズの流れデータ構造で記憶する上記段階は、ディスク記憶装置において関連データを保持する流れデータ構造に対するインデックスを第２の可変サイズの流れデータ構造で記憶する段階を更に備えた請求項１に記載の方法。

【請求項３】　ディスク記憶装置において関連データを保持する流れデータ構造に対するインデックスを第２の可変サイズのデータ構造で記憶する上記段階は、ディスク記憶装置において関連データを保持する流れデータ構造に対するＢツリーインデックスを第２の可変サイズの流れデータ構造で記憶する段階を更に備えた請求項２に記載の方法。

【請求項４】　流れデータ構造の各々に対し流れ記述子を記憶する上記段階は、流れデータ構造の各々に対する流れ記述子を第１及び第２の各々の可変サイズの流れデータ構造で記憶する段階を更に備えた請求項１に記載の方法。

【請求項５】　ディスク記憶装置を有するデータ処理システムにおいて、

（ａ）ディスク記憶装置において論理的に隣接するデータバイトより成る流れデータ構造でデータの流れを記憶し、

（ｂ）ディスク記憶装置において各流れデータ構造に対する流れ記述子を記憶し、各流れ記述子は、流れデータ構造がディスク記憶装置にいかに記憶されるかを記述するタイプ識別子を含み、そして

（ｃ）関連データを記憶する流れデータ構造及びそれらに組み合わされる流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶する、という段階を備えたことを特徴とする方法。

【請求項６】　ディスク記憶装置において流れデータ構造でデータの流れを記憶する上記段階は、ディスク記憶装置において流れデータ構造でメタデータの少なくとも１つの流れを記憶し、そしてディスク記憶装置において流れデータ構造でメタデータではないデータの少なくとも１つの流れを記憶するという段階を更に備えた請求項５に記載の方法。

【請求項７】　各流れデータ構造に対する流れ記述子をディスク記憶装置に記憶する上記段階は、各流れデータ構造に対する流れ記述子をディスク記憶装置において別の流れデータ構造で記憶するという段階を更に備えた請求項５に記載の方法。

【請求項８】　関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶する上記段階は、ファイルのデータを記憶する流れデータ構造及びそれらに組み合わされる流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶するという段階を更に備えた請求項５に記載の方法。

【請求項９】　関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶する上記段階は、ディレクトリーのデータを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶するという段階を更に備えた請求項５に記載の方法。

【請求項１０】　関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶する上記段階は、サブディレクトリーのデータを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶するという段階を更に備えた請求項５に記載の方法。

【請求項１１】　第１の可変サイズのデータ構造で記憶されず、関連データを記憶する流れデータ構造と、それに組み合わされた流れ記述子とを、ディスク記憶装置において第２の可変サイズのデータ構造で記憶するという段階を更に備えた請求項５に記載の方法。

【請求項１２】　第１の可変サイズのデータ構造及び第２の可変サイズのデータ構造をディスク記憶装置においてディスクスペースの固定サイズバケットのアレーに記憶するという段階を更に備えた請求項１１に記載の方法。

【請求項１３】　ディスク記憶装置におけるディスクスペースの固定サイズのバケットのアレーを流れデータ構造で記憶するという段階を更に備えた請求項１２に記載の方法。

【請求項１４】　ディスク記憶装置を有するデータ処理システムにおいて、

（ａ）データの流れをディスク記憶装置において流れデータ構造で記憶し、上記流れは、論理的に隣接するデータバイトより成り、

（ｂ）各流れデータ構造に対する流れ記述子をディスク記憶装置に記憶し、各流れ記述子は、流れデータ構造がディスク記憶装置にいかに記憶されるかを記述するタイ

プ識別子を含み、
（ｃ）関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶し、そして
（ｄ）関連データを記憶する可変サイズのデータ構造のグループをカタログデータ構造に記憶する、という段階を備えたことを特徴とする方法。
【請求項１５】　ディスク記憶装置において流れデータ構造でデータの流れを記憶する上記段階は、ディスク記憶装置において流れデータ構造でメタデータの少なくとも１つの流れを記憶し、そしてディスク記憶装置において流れデータ構造でメタデータではないデータの少なくとも１つの流れを記憶するという段階を更に備えた請求項１４に記載の方法。
【請求項１６】　各流れデータ構造に対する流れ記述子をディスク記憶装置に記憶する上記段階は、各流れデータ構造に対する流れ記述子をディスク記憶装置において別の流れデータ構造で記憶するという段階を更に備えた請求項１４に記載の方法。
【請求項１７】　関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶する上記段階は、ファイルのデータを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶するという段階を更に備えた請求項１４に記載の方法。
【請求項１８】　関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶する上記段階は、ディレクトリーのデータを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶するという段階を更に備えた請求項１４に記載の方法。
【請求項１９】　関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において可変サイズのデータ構造で記憶する上記段階は、サブディレクトリーのデータを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において第１の可変サイズのデータ構造で記憶するという段階を更に備えた請求項１４に記載の方法。
【請求項２０】　第１の可変サイズのデータ構造に記憶されず、関連データを記憶する流れデータ構造と、それに組み合わされた流れ記述子とを、ディスク記憶装置において第２の可変サイズのデータ構造で記憶するという段階を更に備えた請求項１４に記載の方法。
【請求項２１】　第１の可変サイズのデータ構造及び第２の可変サイズのデータ構造をディスク記憶装置においてディスクスペースの固定サイズバケットのアレーに記憶するという段階を更に備えた請求項２０に記載の方法。
【請求項２２】　ディスク記憶装置におけるディスクスペースの固定サイズのバケットのアレーを流れデータ構造で記憶するという段階を更に備えた請求項２１に記載の方法。
【請求項２３】　ディスクスペースの固定サイズバケットのアレーを記憶する流れデータ構造に対する流れ記述子を、関連データを記憶する可変サイズデータ構造の１つに記憶するという段階を更に備えた請求項２２に記載の方法。
【請求項２４】　ディスクスペースの固定サイズバケットのアレーの所定の固定サイズバケットにカタログデータ構造を記憶する段階を更に備えた請求項２２に記載の方法。
【請求項２５】　ディスク記憶装置を有するデータ処理システムにおいて、
（ａ）データの流れをディスク記憶装置において流れデータ構造で記憶し、上記流れは、論理的に隣接するデータバイトより成り、
（ｂ）各流れデータ構造に対する流れ記述子をディスク記憶装置に記憶し、
（ｃ）関連データを記憶する流れデータ構造及びそれらに組み合わされた流れ記述子をディスク記憶装置において各可変サイズのデータ構造で記憶し、各可変サイズのデータ構造は、それに組み合わされた識別子を有し、
（ｄ）組み合わされた識別子を有する可変サイズのデータ構造をディスク記憶装置においてディスクスペースの固定サイズバケットのアレーに記憶し、
（ｅ）上記アレーにおけるバケットに対するバケット識別子を指定するエントリーを保持するマッピング構造をディスク記憶装置に記憶し、上記エントリーは上記可変サイズのデータ構造の識別子によってインデックスされ、そして
（ｆ）上記マッピング構造を使用し、アレーにおける上記可変サイズのデータ構造の１つを、その識別子が与えられると、位置決めする、という段階を備えたことを特徴とする方法。
【請求項２６】　上記アレーは、ディスク記憶装置において流れデータ構造の１つで記憶され、そして上記アレーを記憶する流れに対する流れ記述子がディスク記憶装置に記憶される請求項２５に記載の方法。
【請求項２７】　上記マッピング構造は、ディスク記憶装置において流れデータ構造の１つで記憶され、そして上記マッピング構造を記憶する流れに対する流れ記述子がディスク記憶装置に記憶される請求項２５に記載の方法。
【請求項２８】　上記アレーを記憶する流れに対する流れ記述子及び上記マッピング構造を記憶する流れに対する流れ記述子は、可変サイズのデータ構造の選択された１つで記憶される請求項２７に記載の方法。

【請求項２９】　上記選択された可変サイズのデータ構造は、上記アレーのバケットの１つに記憶される請求項２７に記載の方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は一般にデータ処理システムに係り、より詳細には、ファイルシステムによりディスクにデータを記憶する方法に係る。

【０００２】

【従来の技術】従来のファイルシステムは、ファイルデータをディスクに効率的に記憶する点で問題がある。従来の多くのシステムは、全てのデータをディスクに単一サイズの記憶単位で記憶する手法を採用している。不都合なことに、この手法は、ディスクにファイルデータを効率良く記憶するものではない。特に、ファイルデータはサイズが変化し、従って、所定の記憶単位サイズに良好に一致するものではない。従来の他のシステムは、多数の異なるフォーマットの１つを採用するオプションをユーザに与えるものである。ファイルデータがユーザに使用できるようになる前に、どのフォーマットを採用するか判断しなければならない。その結果、ユーザによるフォーマットの選択は単なる推測となり、実際のファイルデータとうまく対応しないことがしばしばある。従って、ファイルデータが効率的に記憶されないことが頻繁である。

【０００３】

【発明が解決しようとする課題】従来のオペレーティングシステムでは、各ファイルに固定数のディスクスペースのブロックが割り当てられ、ファイルデータと、ファイルに関する制御情報を記憶する。これらのブロックは、単位を素早く割り当てたり割り当て解除したりできるように固定サイズにされている。ファイル及びファイルデータに対する制御情報は、しばしば可変サイズとされ、これが少なくとも２つの問題を課することになる。第１に、ファイルデータ及び／又は制御情報が１つのブロックを埋めるに充分なほど大きくないときには、割り当て単位内のディスクスペースが浪費される。第２に、ファイルデータ及び／又は制御情報が１つのブロックに記憶するには大き過ぎるときには、それが多数のブロックに記憶されねばならず、データがどこに記憶されるかを指定するために多数のポインタが維持されねばならない。このようなポインタを維持しそして使用することは、かなり厄介であることが多い。

【０００４】従来のファイルシステムでは、データとメタデータ（即ち、他のデータの記憶を記述するデータ）とが個別のエントリーとして処理されている。特に、データとメタデータは、従来のファイルシステムでは異なるフォーマットで記憶されている。更に、データ及びメタデータに対して作用する個別のツールが設けられている。データとメタデータをこのように分けて考える結果、オーバーヘッド及び複雑さが増大している。

【０００５】

【課題を解決するための手段】本発明の第１の特徴によれば、ディスク記憶装置と、オペレーティングシステムを実行する処理手段とを有するデータ処理システムにおける方法が実施される。この方法において、データは、ディスク記憶装置に、第１の可変サイズの流れデータ構造で記憶される。メタデータは、ディスク記憶装置に、第２の可変サイズの流れデータ構造で記憶される。各々の流れデータ構造に対して流れ記述子が記憶される。この流れ記述子は、流れデータ構造がディスク記憶装置内のディスクにいかに記憶されるかを識別するタイプ識別子を含んでいる。

【０００６】本発明の別の特徴によれば、データの流れは、ディスク記憶装置に流れデータ構造で記憶される。データの流れは、論理的に隣接するデータバイトから形成される。各流れデータ構造に対しディスク記憶装置に流れ記述子が記憶される。各流れ記述子は、流れデータ構造がディスク記憶装置にいかに記憶されるかを記述するタイプ識別子を含んでいる。関連データを記憶する流れデータ構造は、それらに組み合わされた流れ記述子と共に、第１の可変サイズのデータ構造に記憶される。

【０００７】本発明の更に別の特徴によれば、データの流れは、ディスク記憶装置に流れデータ構造で記憶される。各流れデータ構造ごとに流れ記述子がディスク記憶装置に記憶される。各流れ記述子は、流れデータ構造がディスク記憶装置にいかに記憶されるかを記述するタイプ識別子を含む。関連データを記憶する流れデータ構造は、それに組み合わされた流れ記述子と共に、ディスク記憶装置に可変サイズのデータ構造で記憶される。関連データを記憶する可変サイズのデータ構造のグループは、カタログデータ構造で記憶される。

【０００８】本発明の更に別の特徴によれば、ディスク記憶装置を有するデータ処理システムにおける方法が実施される。この方法では、データの流れはディスク記憶装置に流れデータ構造で記憶される。流れは、論理的に隣接するデータバイトによって形成される。各流れデータ構造ごとに流れ記述子がディスク記憶装置に記憶される。関連データを記憶する流れデータ構造は、それらに組み合わされた流れ記述子と共に、ディスク記憶装置に各々可変サイズのデータ構造で記憶される。各可変サイズのデータ構造は、それに組み合わされた識別子を有している。可変サイズのデータ構造は、ディスク記憶装置におけるディスクスペースの固定サイズの識別可能なバケットのアレーに記憶される。このアレー内のバケットに対するバケット識別子を指定するエントリーを保持するマッピング構造がディスク記憶装置に記憶される。上記エントリーは、可変サイズデータ構造の識別子によってインデックスされる。このマッピング構造は、その識別子が与えられると、上記アレー内の可変サイズデータ

構造の１つを位置決めするのに使用される。

【０００９】

【実施例】本発明の好ましい実施例は、データとメタデータの両方をディスクに「流れ」のグループとして記憶するファイルシステムを提供する。「流れ」とは、データのバイトの論理的に隣接しランダムにアクセスできる可変サイズのアレーであって、ディスク上の論理記憶単位として働くものをいう。ファイルのデータへのほとんどのプログラムアクセスは、これらの流れによって行われる。各流れは、多数の異なる表示(representation)の１つにおいてディスクに記憶される。異なる表示の各々は、流れの特定のサイズ及び使用に良く適したものである。従って、各流れは、そのサイズに最も良く適した表示でディスクに記憶される。

【００１０】各流れには流れ記述子が組み合わされ、これはファイルシステムに制御構造において記憶される。流れ記述子は、流れをアクセスしそして流れに関する情報を得るのに使用される。流れ記述子は、流れのデータを記憶する表示の記述を与える。関連データを保持するデータの流れは、オノード(onode) として知られているデータ構造へとカプセル化される。オノードとは、ファイル、ディレクトリー又はサブディレクトリーにほぼ類似した可変サイズの構造である。関連オノードのグループは、次いで、オブジェクト記憶カタログ又はオブジェクト記憶として知られているデータ構造に記憶される。従って、データ及びメタデータ（オブジェクト記憶カタログのような）は、同様の形態でハイアラーキに記述され、これについては以下で詳細に説明する。

【００１１】図１は、本発明の好ましい実施例によるデータ処理システム１０のブロック図である。図１のシステム１０は単一プロセッサシステムであるが、本発明は分散型システムのようなマルチプロセッサシステムでも実施できることが当業者に明らかであろう。データ処理システム１０は、中央処理ユニット（ＣＰＵ）１２と、メモリ１４と、ディスク記憶装置１６と、キーボード１８と、マウス２０と、ビデオディスプレイ２２とを備えている。ディスク記憶装置１６は、ハードディスク及び他の形式のディスク記憶装置を含む。キーボード１８、マウス２０及びビデオディスプレイ２２は、従来型の入力／出力装置である。

【００１２】メモリ１４は、オペレーティングシステム２４のコピーを保持し、これはシステムに記憶されたファイルを管理するためのファイルシステムマネージャ２６を含んでいる。オペレーティングシステム２４は、オブジェクト指向のオペレーティングシステムである。ここに述べる本発明の好ましい実施例は、オペレーティングシステム２４の一部分として実施される。本発明の好ましい実施例は、オペレーティングシステム２４の一部分として説明するが、当業者であれば、本発明は、オペレーティングシステムとは別の他の形式のコードでも実施できることが明らかであろう。

【００１３】上記したように、本発明の好ましい実施例においては、流れは、多数の異なる表示で得られる。流れについての異なる表示を理解するために、流れに対して与えられる流れ記述子のフォーマットを検討することが有用であろう。図２は、流れ記述子２８のフォーマットを示す図である。流れ記述子２８は、本発明の好ましい実施例で使用できる流れの異なる表示の各々を記述することができる。流れ記述子２８は、サイズフィールド３０、タイプフィールド３２及び記述フィールド３４の３つのフィールドを含んでいる。サイズフィールド３０は、流れのサイズをバイトで指定する値を保持する。タイプフィールド３２は、流れのタイプを指定し、そして記述フィールド３４は、流れの記述（即ち、記述フィールド３４の形式）を保持する。これらフィールド３０、３２及び３４に保持された値は、それに関連する流れの表示と共に変化する（以下で詳細に述べる）。

【００１４】「極小の流れ」は、本発明の好ましい実施例に使用できる流れの第１の表示である。この極小の流れは、記憶媒体（即ち、ディスク記憶装置１６のディスク）の割り当て単位に対して非常にサイズの小さいデータを記憶するのに使用されるものである。記憶媒体の「割り当て単位」とは、ファイルを記憶するために割り当てられるディスク記憶装置１６のディスクメモリスペースの基本単位を指す。例えば、ＦＡＴベースのファイルシステムでは、最小割り当て単位はディスクセクタである。不都合なことに、このセクタは、システム１０で形成される流れデータよりも相当に大きいことがしばしばある。図３は、極小の流れに対する流れ記述子２８のフォーマットを示している。サイズフィールド３０は流れのサイズを指定する値を保持し、そしてタイプフィールド３２は、流れが極小の流れであることを指定する。記述フィールド３４は、流れのデータを保持し、従って、流れのデータの即時表示を与える。この即時表示は、少量のデータを記憶するための非常に効率的な手段を形成する。特に、データは、流れ記述子に直接的に合体されて、容易に且つ速やかにアクセスできるようにされる。

【００１５】本発明の好ましい実施例に使用できる別の表示は、小さな流れである。「小さな流れ」とは、データの単一のイクステントで記憶される流れである。イクステントとは、割り当て単位の可変サイズの隣接した延びである。流れのデータは、流れ記述子に直接記憶するには大き過ぎるのでイクステントに記憶される。この小さな流れに対する流れ記述子２８のフォーマットが図４に示されている。タイプフィールド３２は、それに関連した流れが小さな流れであることを指定し、そして記述フィールド３４は、流れのデータが記憶されるイクステント４２を記述するイクステント記述子３６を保持する。イクステント４２は、ディスク記憶装置１６のディ

スクに記憶される。イクステント記述子３６は、２つのサブフィールド３８及び４０を含む。サブフィールド３８はイクステント４２の長さを指定する値を保持し、そしてサブフィールド４０は、イクステントのディスクアドレス（即ち、イクステントがディスクの論理アドレススペースのどこに位置しているか）を保持する。

【００１６】本発明の好ましい実施例に使用できる第３の表示は、「大きな流れ」である。この大きな流れは、多数のイクステントに記憶される流れである。この大きな流れは、多量のデータを有する流れを記憶するのに適している。図５は、このような大きな流れに対する流れ記述子２８のフォーマットを示している。タイプフィールド３２は、流れが大きな流れであることを指定する。記述フィールド３４は、イクステント記述子を保持する流れ４４を記述する第２の流れ記述子４３を保持する。この第２の流れ記述子４３は極小の流れを記述し、そしてイクステント記述子の流れ４４を保持する記述フィールド３４’を含んでいる。イクステント記述子３６’、３６”及び３６”’は、図４について述べたイクステント記述子３６と同じフォーマットを有する。その結果、単一の流れ３４’によって多数のイクステント４２’、４２”及び４２”’が記述される。イクステント記述子の数があまりに多過ぎる場合には、第２の流れ記述子４３は、極小の流れではなくて小さな流れを記述する。更に、イクステント記述子の数が小さな流れに対して多過ぎる場合には、第２の流れ記述子４３は、大きな流れを記述する。大きな流れは、一般に、ディスクスペースの大きな隣接ブロックが得られない場合に使用される。この大きな流れは、多量のデータをディスクに対して分散されたイクステントにおいて単一の流れとして容易に記憶するものである。その結果、大きな流れは、ディスクが更に細分化されそして流れが成長するときにも、良好にスケーリングされる。

【００１７】流れのサイズが成長するときには、流れの効率的な記憶を容易にするために、流れの表示が流れ表示のハイアラーキを登るように促される。ハイアラーキは、極小の流れ、小さな流れ及び大きな流れを含む。流れは、極小の流れから小さな流れへそして大きな流れへと促進される。一般に、上記したように、流れに含まれたデータの量及び細分の量に基づいて流れに最も適した表示が選択される。

【００１８】本発明の好ましい実施例に使用できる４つの基本的な流れの形式を以上に説明した。流れ記述子２８のタイプフィールド３４は、流れ内に記憶されるデータの特殊な記述を指定するのにも使用できる。図６は、流れが圧縮データを保持するときの流れ記述子２８のフォーマットの一例である。タイプフィールド３２は、流れのデータが圧縮されることを指定する値を保持し、一方、記述フィールド３４は、圧縮データに対する流れ記述子を保持する。記述フィールド３４に保持される流れ記述子は、流れに含まれるデータの量に基づいて、極小の流れ、小さな流れ、又は大きな流れである。

【００１９】図７は、流れ記述子が暗号データの流れを記述するときの流れ記述子２８のフォーマットを示している。タイプフィールド３２は、流れが暗号データを保持することを指定する値を保持する。記述フィールド３４は、暗号データのための流れ記述子を保持する。又、流れ記述子は、暗号キーの値５０も含む。この暗号キーの値５０は、流れに記憶されたデータを暗号解読するのに使用できる。記述フィールド３４に保持された暗号データに対する流れ記述子は、極小の流れ、小さな流れ、又は大きな流れである。

【００２０】データの特殊な記述を指定するようにタイプフィールド３２を使用する別の例が図８に示されている。図８は、小規模のトランザクションに対する流れ記述子２８を示している。小規模のトランザクションとは、データベースのデータに対する変更が記録されるが、影響を受ける全てのデータの変更に関連したオーバーヘッドを被ることを保証するに充分な数の他の変更が生じるまでデータを直接的に変更しない場合を指す。タイプフィールド３２は、記述フィールド３４が小規模なトランザクションに対するデータを保持することを指定する。記述フィールド３４は、第１の流れ記述子５２と、第２の流れ記述子５４を保持する。第１の流れ記述子５２は、データの元の流れを記述する。第２の流れ記述子５４は、元の流れに対して行われた変更を指定する流れを記述する。データの元の流れは、第２の流れに保持された変更を実施することによって更新される。

【００２１】図９は、複製データを保持する流れに対する流れ記述子２８の例を示す。データのロスが破壊的な結果を招かないように、データをしばしば複製しなければならない。特に、システムが多数のコピーをディスクに維持するところの選択されたデータ構造体がある。このような場合に、データ構造体は、ディスク上の２つの異なる位置にコピーされる。流れ記述子２８は、第１の流れ記述子５６と、第２の流れ記述子５８をその記述フィールド３４に含んでいる。第１の流れ記述子５６は、第１のデータコピーを保持する第１の流れを記述し、そして第２の流れ記述子５８は、別のデータコピーを保持する第２の流れを記述する。タイプフィールド３２は、流れが複製データを含むことを指定する値を保持する。

【００２２】本発明の好ましい実施例では、流れに関する情報が図１０に示すようなフィールド記述子６０に記憶される。このフィールド記述子６０は、流れＩＤを保持する流れＩＤフィールド６２を含んでいる。この流れＩＤは、オノード（以下で詳細に述べる）内の流れを独特に識別する４バイト長さの識別番号である。更に、フィールド記述子６０は、フラグビットを保持するフラグフィールド６４を備えている。フィールド記述子６０の最終フィールドは、流れに対する流れ記述子２８であ

る。

【００２３】流れは、関連ファンクションに基づいて「オノード」にグループ分けされる。オノードはオブジェクトの論理表示に対応し、典型的に、ファイル、ディレクトリー又はサブディレクトリーを構成する全ての流れを保持する。各オノードは、これに含まれる流れの可変サイズの集合体を記述するに必要な情報を含む。

【００２４】図１１は、オノード６６のフォーマットを示す図である。各オノード６６は、関連ファンクションの流れを保持している。各オノード６６は、次のフィールドを含む。即ち、長さフィールド６８と、ワークＩＤフィールド７０と、フラグフィールド７２と、クラスＩＤフィールド７３と、フィールド記述子６０（図１０に示すような）のアレーを保持するフィールド７６とである。長さフィールド６８は、オノードの長さを指定する値を保持し、一方、ワークＩＤフィールド７０は、以下で詳細に述べるように、ワークＩＤマッピングアレー１０４（図１４）へのインデックスを保持する。ワークＩＤは長さが４バイトである。フラグフィールド７２（図１１）はフラグビットを保持し、そしてクラスＩＤフィールド７３は、オノードに対するクラスＩＤを保持する。フィールド７６は、フィールド記述子６０のパックアレーを保持し、これは、オノード６６と共に保持される流れの各々に対するフィールド記述子６０を含む。フィールド７６のフィールド記述子のアレーに含まれる流れの数は、変化し得る。更に、フィールド７６のフィールド記述子のアレーにおける各流れの長さも、変化し得る。従って、オノード６６は可変サイズ構造である。オノード６６の可変サイズ特性は、ディスク記憶装置１６のディスクにおける割り当て単位での内部分割を最小にするよう助成する。

【００２５】オノード６６に関するあるデータは、オノードに直接組み込まれない。そうではなくて、このデータは、図１２に示すように個別の流れに記憶される。流れ７８は、関連オノード６６（図１１）に関する多数の異なる形式の状態情報を保持する。この状態情報は、フィールド８６に保持されたタイムスタンプであって、オノード６６が形成された時間を指定するタイムスタンプを含んでいる。フィールド８８は、オノード６６が変更された最後の時間を指定するタイムスタンプを保持する。同様に、フィールド９０は、オノード６６がアクセスされた最後の時間を指定するタイムスタンプを保持する。フィールド９２は、オノード６６のサイズを指定する値を保持し、そしてフィールド９４は、オノードのオーナーに対する機密記述子を保持する。流れ７８に保持された全ての情報は、関連オノード６６に保持されたファイルデータを管理するのに有用である。

【００２６】図１２には、状態情報の第２の流れ８０も示されている。この流れ８０は、３つのフィールド９６、９８及び１００を備えている。各オノード６６は、データ処理システム１０のグローバルネームスペースに見ることができ、そしてこのグローバルネームスペースは、ルートオノード以外の各オノードが親ノードを有する論理ツリー構造である。フィールド９６は、親オノードのワークＩＤを保持する。フィールド９８は、オノード６６のための全般的に独特のＩＤ（ＵＵＩＤ）を保持する。更に、フィールド１００は、オノード６６のクラスを指定するクラスＩＤを保持する。各オノード６６には独特のクラスが組み合わされる。特に、各オノード６６は、特定のクラスのオブジェクトの例である。

【００２７】状態情報の２つの付加的な流れ８２及び８４も記憶される。流れ８２は、親に対するオノード６６のネームを保持し、そして流れ８４は、オノードのためのアクセス制御リスト（機密で使用される）を保持する。

【００２８】流れ７８、８０、８２及び８４は、少なくとも次の２つの理由で別々に記憶される。第１に、この情報を別々に記憶すると、オノード６６の平均サイズが減少される。流れ７８、８０、８２及び８４は、各オノードごとに記憶されない。その結果、オノードの平均サイズが減少する。第２に、この情報を別々に記憶すると、各オノード６６からコードを検索するのに複雑なコードを必要とすることなく、情報の関連グループにプログラムアクセスすることができる。

【００２９】オノード６６（図１１）は、オノードバケットアレー１０２（図１３）に記憶される。このオノードバケットアレー１０２は、固定サイズバケットのアレーで構成された可変サイズデータ構造である。バケットのサイズ（例えば、４Ｋ）はデータ処理システム１０のアーキテクチャに基づくものであるが、システム１０のページサイズに一致してもよい。これらバケットは、図１３の例では１ないしＮと番号付けされている。アレー１０２の各バケットは、オノード６６のパックされた組を含んでいる。図１３の例では、バケット２は、単一のオノードのみを含み、一方、バケット１は、２つのオノードを含み、そしてバケット３は、３つのオノードを含むことが明らかである。

【００３０】オノードバケットアレー１０２は、ファイルデータが移動、削除又は挿入されたときにシャフルされねばならないデータの量を最小にするために使用される。ディスクスペースのブロックを割り当て及び割り当て解除する粒度は固定とされる。換言すれば、メモリブロックは、固定サイズのバケットで割り当て及び割り当て解除される。そうではなくて、ファイルデータを記憶するために可変サイズの構造が使用された場合には、割り当ての粒度は固定されず、割り当ての粒度を非常に大きくすることができる。

【００３１】効率化のために、オペレーティングシステム２４（図1）は、関連オノード６６（図１１）をアレー１０２（図１４）の同じバケットに記憶するか、又は

ディスク記憶装置１６のディスク上で互いに接近したバケットに記憶する。この記憶戦術は、ディスク上でオノード６６を探すシーク時間を最小にする。一般に、通常一緒にアクセスされるファイルは同じバケットに配置される。一緒にアクセスされるファイルは、例えば、共通のサブディレクトリー又はディレクトリーにあるファイルである。

【００３２】オノードバケットアレー１０２（図１３）は、流れとしてアクセスされる。システム１０は、通常、多数のオノードバケットアレー１０２を記憶し、従って、オノードバケットアレーを保持する多数の流れを記憶する。内部では、オノード間の全ての参照は、ワークＩＤに基づいている。アレー１０２のバケット内でオノード６６（図１１）を位置決めするには、そのオノードに対するワークＩＤを探すことが必要である。ワークＩＤのマッピングアレー１０４（図１４）は、ワークＩＤからオノードバケットアレー１０２のバケット番号へマップする。次いで、番号付けされたバケットがオノードに対し整合ワークＩＤでサーチされる。特に、特定ノード６６のワークＩＤは、ワークＩＤマッピングアレー１０４へインデックスを与える。指定されたインデックスにおけるエントリーは、オノード６６を保持するバケット番号を識別する。例えば、図１４に示すように、ワークＩＤマッピングアレー１０４のエントリー１０６及び１０８は、関連するワークＩＤを有するオノード６６がバケット１に保持されることを指定する値を保持する。

【００３３】各オノード６６（図１１）は、フィールド記述子７６のアレーに保持される流れの中に記憶されるネームインデックスの流れを含んでいる。ネームインデックスの流れは、オノード６６内に含まれた流れ記述子に対するＢツリーインデックスを保持する。ネームインデックスの流れは、流れの流れＩＤを、オノード６６内に保持された流れ記述子を位置決めするためのキーとして使用する。Ｂツリーインデックスは、本発明と同日に出願された共通の譲受人に譲渡された「ファイルシステムにオブジェクトを効率的に記憶する方法(Efficient Storage of Objects in a File System)」と題する特許出願に詳細に示されている。この特許出願の開示を参考としてここに取り上げる。

【００３４】流れの関連収集体がオノード６６（図１１）へと収集されるのと同様に、オノードの関連収集体がオブジェクト記憶カタログとして知られたデータ構造へ収集される。オノード６６のワークＩＤは、オブジェクト記憶カタログ内のオノードを識別するためのベースとして働く。オブジェクト記憶カタログは、図１５に示すようなオブジェクト記憶カタログオノード１１０によって記述される。オブジェクト記憶カタログオノード１１０は、長さフィールド６８と、ワークＩＤフィールド７０と、フラグフィールド７２と、他のオノード６６で見つかったフィールド記述子７６のアレーとを含んでいる。フィールド記述子７６のアレーは、流れ記述子１１２、１１４及び１１６が記憶されたフィールド記述子を含む。

【００３５】流れ記述子１１２は、ワークＩＤマッピングアレーの流れ１０４を記述する。流れ記述子１１４は、オノードバケットアレーの流れ１０２を記述し、そして流れ記述子１１６は、ネームインデックスの流れ１１７を記述する。ワークＩＤマッピングアレーの流れ１０４及びオノードバケットアレーの流れ１０２は、関連データを記憶するので、これらは共通のオノード（即ち、オブジェクト記憶カタログオノード１１０）に記憶される。このように、オブジェクト記憶カタログオノード１１０は、ワークＩＤマッピングアレーの流れ１０４、オノードバケットアレーの流れ１０２及びネームインデックスの流れ１１７を関係付けるピークルを形成する。オブジェクト記憶カタログオノード１１０は、ワークＩＤ　０におけるカタログを構成するオノードのグループ内に記憶される。ワークＩＤ　０とは、その定義により、オノードバケットアレーの流れ１０２の第１エレメントに常に配置される。ワークＩＤ　０における第１バケットは、既知の位置を有する区別されたバケットである。従って、バケット内に記憶されたデータ構造は容易に且つ簡単にアクセスされる。

【００３６】本発明を好ましい実施例について説明したが、特許請求の範囲に規定された本発明の範囲から逸脱せずにその形態及び細部に種々の変更がなされ得ることが当業者に明らかであろう。

【図面の簡単な説明】

【図１】本発明の好ましい実施例によるデータ処理システムのブロック図である。

【図２】本発明の好ましい実施例に用いられる流れ記述子のフォーマットを示す図である。

【図３】本発明の好ましい実施例による極小の流れに対する流れ記述子の図である。

【図４】本発明の好ましい実施例による小さな流れに対する流れ記述子の図である。

【図５】本発明の好ましい実施例による大きな流れに対する流れ記述子の図である。

【図６】本発明の好ましい実施例による圧縮流に対する流れ記述子の図である。

【図７】本発明の好ましい実施例による暗号流に対する流れ記述子の図である。

【図８】本発明の好ましい実施例による小規模トランザクションに対する流れ記述子の図である。

【図９】本発明の好ましい実施例による複製データに対する流れ記述子の図である。

【図１０】本発明の好ましい実施例によるフィールド記述子の図である。

【図１１】本発明の好ましい実施例によるオノードの図

である。

【図１２】図１１のオノードについての適切な情報を保持するシステムの図である。

【図１３】本発明の好ましい実施例によるバケットアレーの図である。

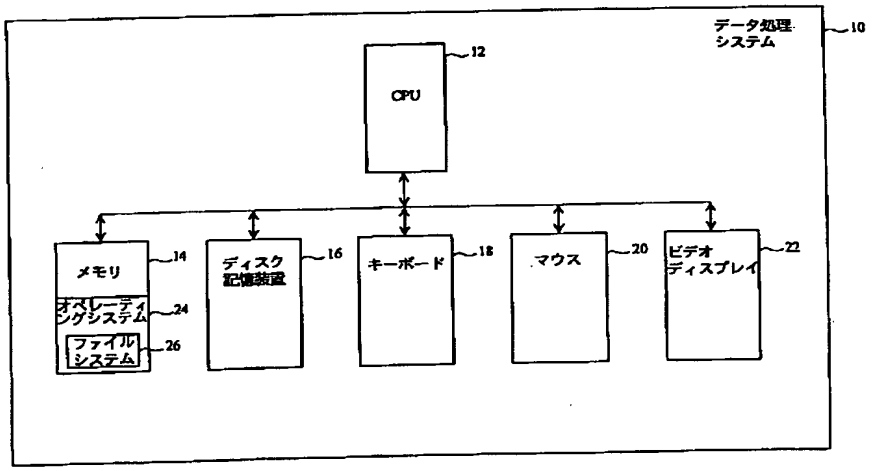
【図１４】本発明の好ましい実施例によるワークＩＤマッピングアレー及びオノードバケットアレーの図である。

【図１５】本発明の好ましい実施例によるオブジェクト記憶カタログの図である。

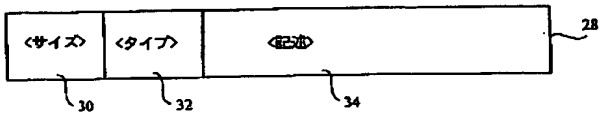
【符号の説明】

１０　データ処理システム
１２　中央処理ユニット
１４　メモリ
１６　ディスク記憶装置
１８　キーボード
２０　マウス
２２　ビデオディスプレイ
２４　オペレーティングシステム
２６　ファイルシステム
２８　流れ記述子
３０　サイズフィールド
３２　タイプフィールド
３４　記述フィールド
３８、４０　サブフィールド
４２　イクステント
４３　第２の流れ記述子
６０　フィールド記述子
６６　オノード
６８　長さフィールド
７０　ワークＩＤフィールド
７２　フラグフィールド
７３　クラスＩＤフィールド
７６　フィールド
７８　流れ
８０　第２の流れ

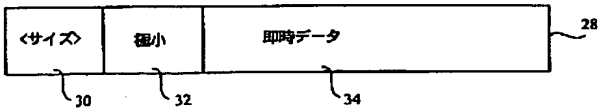
【図１】



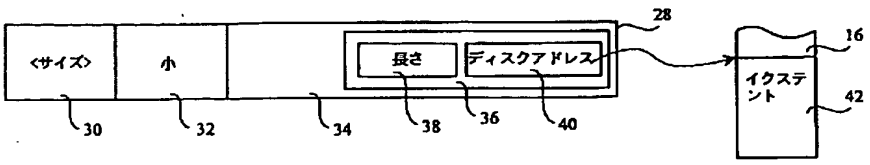
【図２】



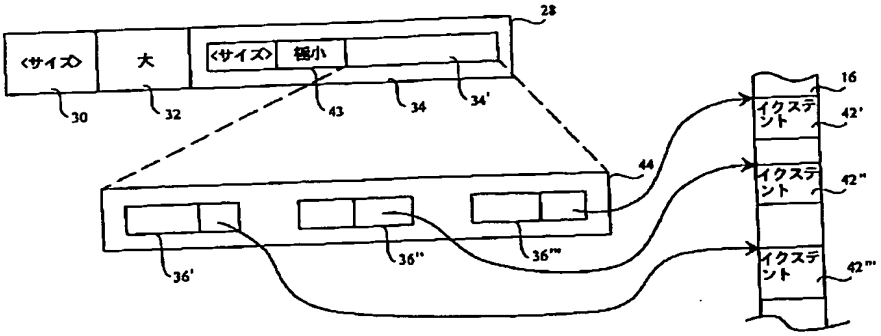
【図３】



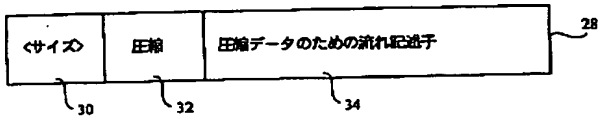
【図４】



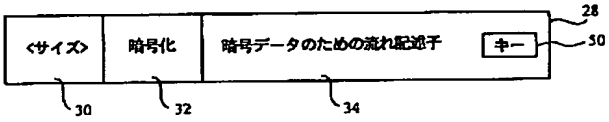
【図５】



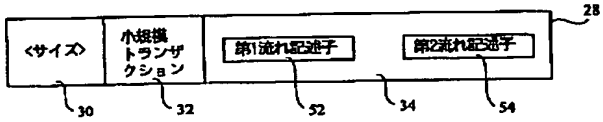
【図６】



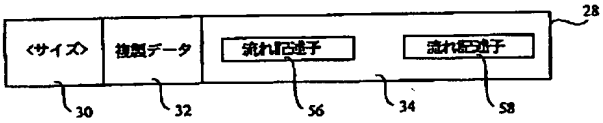
【図７】



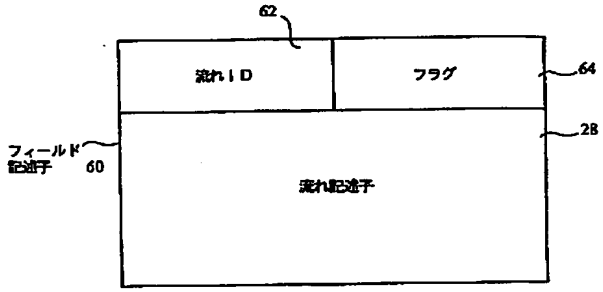
【図８】



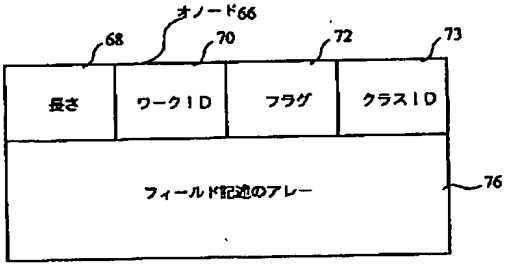
【図９】



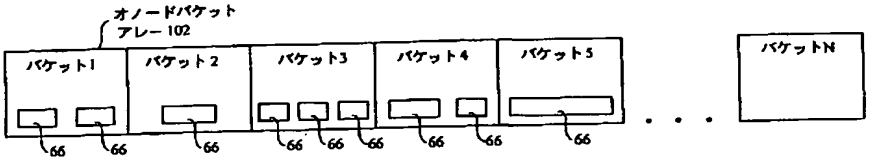
【図１０】



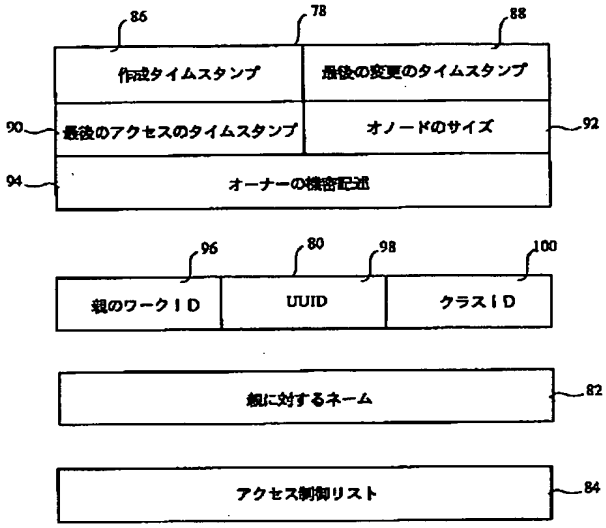
【図１１】



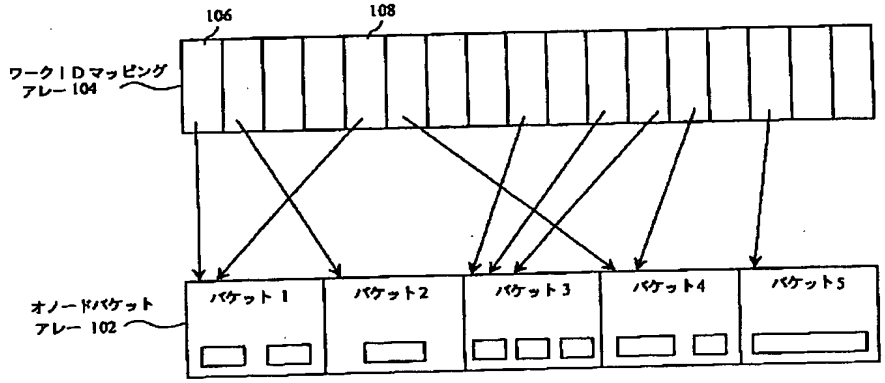
【図１３】



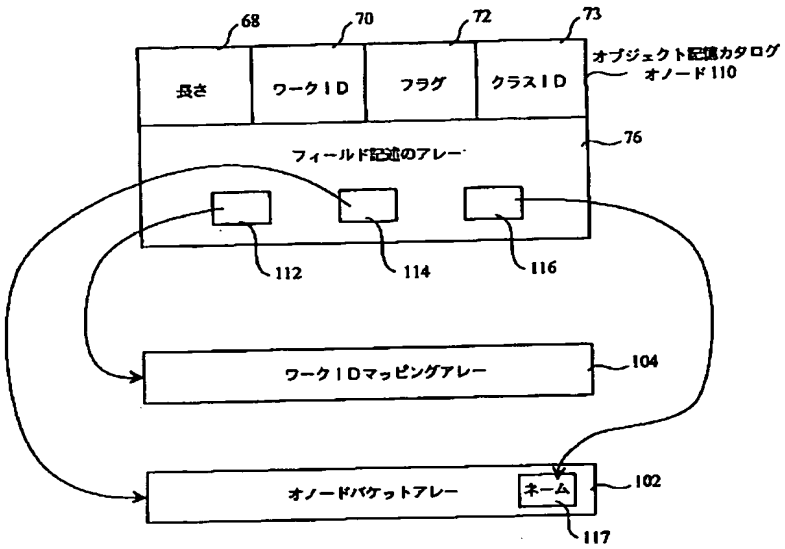
【図１２】


86
78
88
作成タイムスタンプ
最後の変更のタイムスタンプ
90
最後のアクセスのタイムスタンプ
オノードのサイズ
92
94
オーナーの機密記述
96
80
98
100
親のワークＩＤ
UUID
クラスＩＤ
親に対するネーム
82
アクセス制御リスト
84

【図１４】


106
108
ワークＩＤマッピング
アレー 104
オノードバケット
アレー 102
バケット１
バケット２
バケット３
バケット４
バケット５

【図１５】



フロントページの続き

(72)発明者　ブライアン　ティー　バーコウィッツ
アメリカ合衆国　ワシントン州　98007
ベルヴィユー　ワンハンドレッドアンドフォーティセカンド　プレイス　ノース　イースト　3912

(72)発明者　ロバート　アイ　ファーガソン
アメリカ合衆国　ワシントン州　98119
シアトル　ナインス　アベニュー　ウェスト　2910